Interactive visual data exploration with subjective feedback : an information-theoretic approach

Puolamäki, Kai

2020-01

Puolamäki , K , Oikarinen , E , Kang , B , Lijffijt , J & Bie , T D 2020 , ' Interactive visual data exploration with subjective feedback : an information-theoretic approach ' , Data Mining and þÿ K n o w l e d g e D i s c o v e r y , v o l . 3 4 , n o . 1 , p p . 2 1 4 9 . h t t p s : / / d o i . o r g / 1 0 . 1

http://hdl.handle.net/10138/317145
https://doi.org/10.1007/s10618-019-00655-x

Data Mining and Knowledge Discovery (2020) 34:21–49
https://doi.org/10.1007/s10618-019-00655-x

Interactive visual data exploration with subjective feedback: an information-theoretic approach

Kai Puolamäki[1] · Emilia Oikarinen[1] · Bo Kang[2] · Jefrey Lijffijt[2] · Tijl De Bie[2]

Received: 7 August 2018 / Accepted: 13 September 2019 / Published online: 3 October 2019

Abstract

Visual exploration of high-dimensional real-valued datasets is a fundamental task in exploratory data analysis (EDA). Existing projection methods for data visualization use predefined criteria to choose the representation of data. There is a lack of methods that (i) use information on what the user has learned from the data and (ii) show patterns that she does not know yet. We construct a theoretical model where identified patterns can be input as knowledge to the system. The knowledge syntax here is intuitive, such as "this set of points forms a cluster", and requires no knowledge of maths. This background knowledge is used to find a maximum entropy distribution of the data, after which the user is provided with data projections for which the data and the maximum entropy distribution differ the most, hence showing the user aspects of data that are maximally informative given the background knowledge. We study the computational performance of our model and present use cases on synthetic and real data. We find that the model allows the user to learn information efficiently from various data sources and works sufficiently fast in practice. In addition, we provide an open source EDA demonstrator system implementing our model with tailored interactive visualizations. We conclude that the information theoretic approach to EDA where patterns observed by a user are formalized as constraints provides a principled, intuitive, and efficient basis for constructing an EDA system.

Responsible editor: Pauli Miettinen

This work has been supported by the European Research Council (ERC) under the EU's Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement No. 615517, the Research Foundation—Flanders FWO (Project Nos. G091017N, G0F9816N), the EU's Horizon 2020 research and innovation programme and the FWO under the MSC Grant Agreement No. 665501, the Academy of Finland (decisions 326280 and 326339), and Tekes (Revolution of Knowledge Work Project).

✉ Kai Puolamäki
kai.puolamaki@helsinki.fi

[1] Department of Computer Science, University of Helsinki, Helsinki, Finland

[2] Department of Electronics and Information Systems, IDLab, Ghent University, Ghent, Belgium

Keywords Exploratory data analysis · Dimensionality reduction · Information theory · Subjective interestingness · Maximum entropy distribution

1 Introduction

Ever since Tukey's pioneering work on *exploratory data analysis* (EDA) (Tukey 1977), the task of effectively exploring data has remained an art as much as a science. Indeed, while human analysts are remarkably skilled in spotting patterns and relations in adequately visualized data, coming up with insightful visualizations is a hard task to formalize, let alone to automate. As a result, EDA systems require significant expertise to use effectively. However, with the increasing availability and importance of data, data analysts with sufficient expertise are becoming a scarce resource. Thus, further research into automating the search for insightful data visualizations has become increasingly critical.

Modern computational methods for dimensionality reduction, such as Projection Pursuit and manifold learning, allow one to spot complex relations from the data automatically and to present them visually. Their drawback is however that the criteria by which the views are found are defined by *static objective functions*. The resulting visualizations may or may not be informative for the user and task at hand. Often such visualizations show the most prominent features of the data, while the user might be interested in other, perhaps subtler, structures. It would therefore be of a great help if the user could efficiently tell the system what she already knows and the system could utilize this when deciding what to show the user next. Achieving this is the main objective of this paper.

In this paper, we present a novel interactive framework for EDA based on solid theoretical principles and taking into account the updating knowledge of the user. Our work is motivated by the ideas in (Puolamäki et al. 2016; Kang et al. 2016), and in Sect. 4 we discuss in detail the differences between the previous approach and our current one, as well as review several other related approaches, such as iPCA (Jeong et al. 2009), InVis (Paurat and Gärtner 2013), supervised PCA (Barshan et al. 2011), and guided locally linear embedding (Alipanahi and Ghodsi 2011).

Our main idea is shown in Fig. 1. (a) The belief state of the user is modeled using a distribution maintained by the computer (*the background distribution*), which may be partially learned from the data. (b) The system computes a projection (*the most informative projection*) optimizing a user-picked criterion, while factoring out what is already modeled/known about the data. The intuitive idea is that the projection computed shows the maximal difference between the data and the background distribution (i.e., the belief state of the user). (c) The user is shown the data in the most informative projection. (d) The user explores the visualization. (e) The user marks observed clusters on the projection, and (f) the computer then uses these as *constraints* to update the background distribution. The process is iterated until the user is satisfied, i.e., typically when there are no more notable differences between the data and the background distribution.

Specifically, the data considered in this work is a set of d-dimensional (d-D) data points. To illustrate the envisioned data exploration process, we synthesized a 3-D

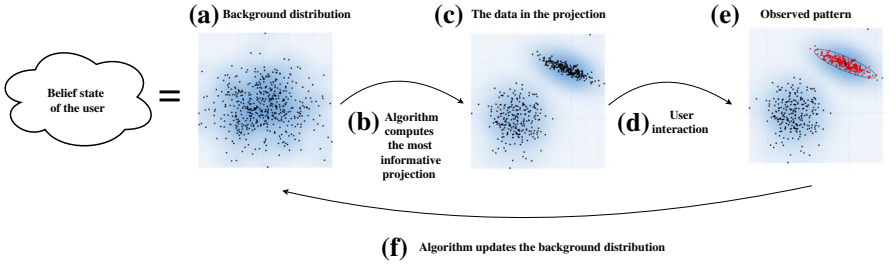

Fig. 1 Overview of the interaction process. **a** The belief state of the user is modeled as the background distribution. **b** The system computes the most informative projection with respect to what is known about the data. **c** The user is shown the data in the most informative projection. **d** The user explores the visualization, **e** marks observed clusters on the projection. **f** The computer uses these as constraints to update the background distribution. The process is iterated until the user is satisfied

dataset with 150 points such that there are two clusters of 50 points and two of 25 points. The smaller clusters are partially overlapping in the third dimension. Looking at the first two principal components, one can only observe three clusters with 50 points each (similarly to the black points in Fig. 2a).

In our iterative approach, the data analyst will learn not only that there are actually four clusters, but also that two of the four clusters correspond to a single cluster in the first view of the data. The visualizations considered are scatterplots of the data points after projection onto a 2-D subspace, as in Projection Pursuit (Friedman and Tukey 1974; Huber 1985). The projection chosen for visualization is the one that, for a specified statistic—the demo system includes variance (PCA) and higher-order moments (ICA)—is maximally different with respect to the background distribution that represents the user's current understanding of the data.

In addition to showing the data in the scatterplot (black points), we display a sample from the background distribution as gray points, see Fig. 2 for an example. The lines shown connect a data point to a respective point in the sample from the background distribution and provide an indication of the displacement in the background distribution for each data point, see Sect. 3.2 for details. The data analyst's interaction consists of informing the system about sets of data points they perceive to form clusters within this scatterplot (Fig. 2b). The information about the user's knowledge of the data is then taken into account and the background distribution is updated accordingly. Figure 2c shows the same projection but a new sample from the background distribution. This sample aligns very well with the observed cluster structure, illustrating the updates to the background distribution were appropriate.

When we have ascertained ourselves that the background distribution matches the data in the current projection as we think it should, the system can be instructed to find another 2-D projection. The new projection displayed is the one that is maximally insightful, *considering the updated background distribution*. We achieve this through the use of a whitening operation (Kessy et al. 2018), which is explained in detail in Sect. 2.5. The underlying idea is that a direction-preserving whitening transformation of the data using the background distribution results in a unit Gaussian spherical distribution, if the data follows the background distribution. Hence, we can use the

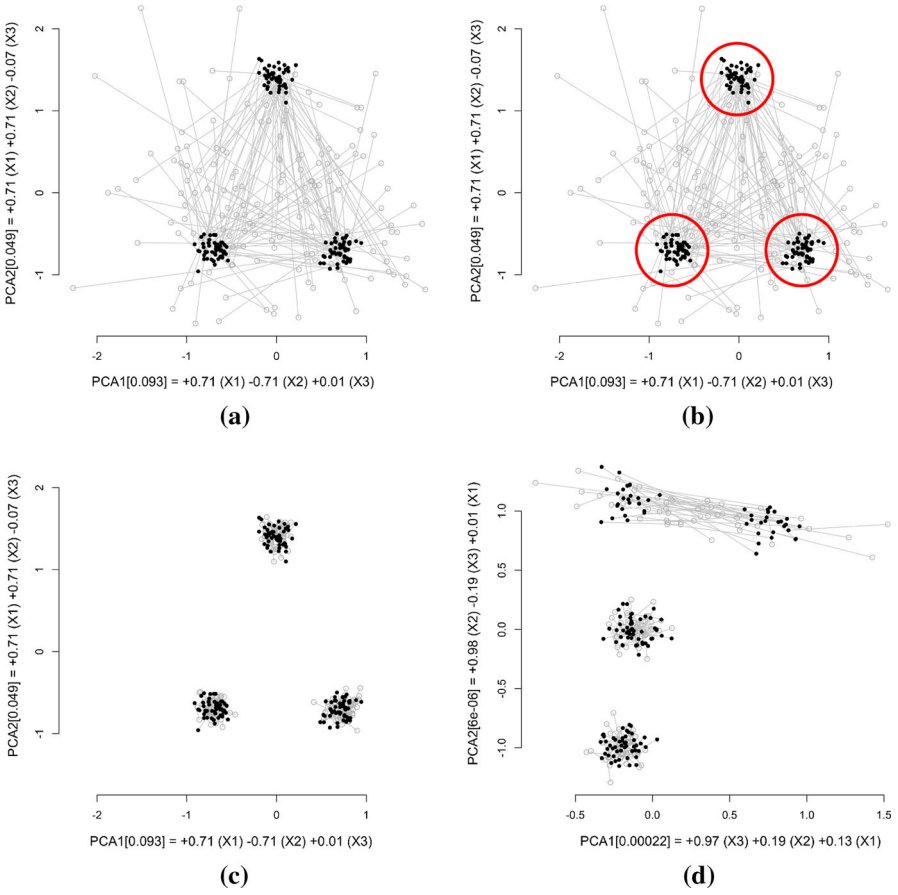

Fig. 2 Synthetic data with 3 dimensions. **a** Projection of the data to the first two principal components together with a sample of background distribution; **b** the user marks the three clusters that she identified; **c** after the user's knowledge is taken into account, the updated background distribution matches the data in this projection; **d** the user is then shown the next most informative projection

deviation from the unit sphere distribution in the whitened data as a signal of difference between the data and the current background distribution. The next projection for our example dataset is shown in Fig. 2d and reveals that one of the three clusters from the previous view can in fact be meaningfully split into two clusters. Notice here that the knowledge learned from the previous view only partially describes the relations in the data, and the new view allows the user to update her knowledge.

The user can now add further knowledge to the background distribution by selecting the two uppermost clusters and the process can be repeated. For our 3-D dataset, after the background distribution is updated upon addition of the new knowledge, the data and the background distribution match, and in this case, further projections will not reveal any additional structure.

Regarding the scope, the focus in this paper is on projection methods that can take into account background knowledge that is updated iteratively when using the system.

There are important open questions regarding the interface and the interaction process, such as how to help (lay) users understand scatterplots based on dimensionality reduction, see, e.g., Sedlmair et al. (2012) for an overview of associated problems and Stahnke et al. (2016) for some recent developments to address some of these. However, the quest to automate the composition of insightful visualizations is important in its own right, as is illustrated in the remainder of the paper.

1.1 Contributions and outline of the paper

The contributions of this paper are:

- A formalization of an efficiently computed background distribution accounting for a user's knowledge in terms of a constrained Maximum Entropy distribution.
- A principled way to obtain projections showing the maximal difference between the data and the background distribution for the principal component analysis (PCA) and independent component analysis (ICA) objectives, by *whitening* the data with respect to the background distribution.
- An interaction model by which the user can input what she has learned from the data in terms of sets of data points (e.g., clusters), which translate into constraints on the background distribution.
- An experimental evaluation of the computational performance of the algorithms used in our framework and use cases on real data.
- A free open source demonstrator system implementing the method.

The current work extends considerably an earlier short poster paper (Puolamäki et al. 2018). In addition to extending the discussion throughout the manuscript, the new material contains (i) a running example using a simple crafted data to illustrate the main concepts of our exploration framework (Examples 1–6); (ii) a detailed description how to efficiently solve Problem 1 (Sect. 2.3.1) (iii) an analysis of the convergence of the optimization (Sect. 2.4); (iv) a section summarizing the exploration flow in our framework (Sect. 2.6); and (v) a discussion of related work (Sect. 4).

The paper is structured as follows: we describe the methods and the algorithms in Sect. 2, report the results of the experiments in Sect. 3 together with a proof-of-concept open source implementation SIDER, discuss the related work in Sect. 4, and finally conclude in Sect. 5.

2 Methods

In this section we present our methods. We start with the preliminaries in Sect. 2.1 and present then our main concepts, namely the constraints and the background distribution in Sect. 2.2. We show how we can update the background distribution in Sect. 2.3 and discuss convergence issues in Sect. 2.4. Finally, we show how to find directions where the data and the background distribution differ using an advanced *whitening operation* in Sect. 2.5 and summarize our framework for interactive visual data exploration in Sect. 2.6.

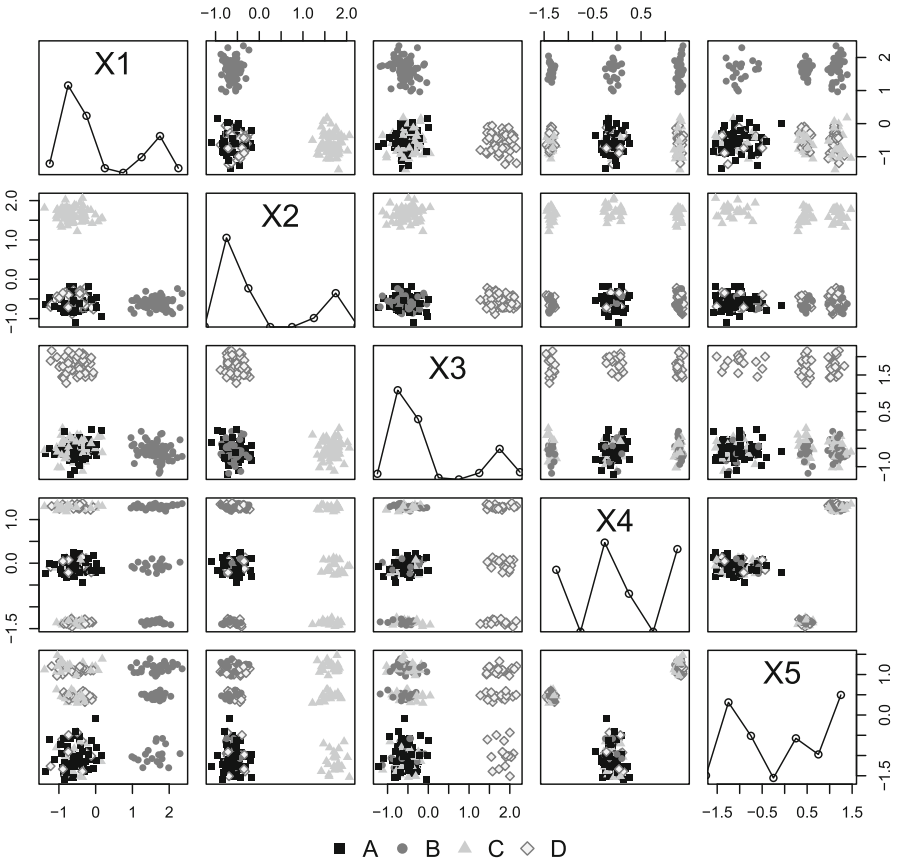

Fig. 3 A pairplot of the synthetic data $\hat{\mathbf{X}}_5$. The point types correspond to the cluster identities A, B, C, and D (the grouping that exists in the first three dimensions). Plot is based on a sample of 250 points from the data

2.1 Preliminaries

We assume the dataset under analysis consists of n d-dimensional real vectors $\hat{\mathbf{x}}_i \in \mathbb{R}^d$, where $i \in [n] = \{1, \ldots, n\}$. The whole dataset is represented by a real-valued matrix $\hat{\mathbf{X}} = \left(\hat{\mathbf{x}}_1 \hat{\mathbf{x}}_2 \ldots \hat{\mathbf{x}}_n\right)^T \in \mathbb{R}^{n \times d}$. We use hatted variables (e.g., $\hat{\mathbf{X}}$) to denote the data and non-hatted variables (e.g., $\mathbf{X}$) to denote the respective random variables.

Example 1 (Running example, see Fig. 3) To illustrate the central concepts of the approach, we generated a synthetic dataset $\hat{\mathbf{X}}_5$ of 1000 data vectors in five dimensions (denoted by X1, ..., X5). The dataset is designed so that along dimensions X1–X3 it can be clustered into four clusters (labeled A, B, C, D) and along dimensions X4 and X5 into three clusters (labeled E, F, G). The clusters in dimensions X1–X3 are located such that in any 2-D projection along these dimensions cluster A overlaps with one of the clusters B, C, or D. The cluster structure in dimensions X4 and X5 is loosely related to the cluster structure in dimensions X1–X3: with 75% probability a

data vector belonging to clusters B, C, or D belongs to one of clusters E and F. The remaining points belong to cluster G. The pairplot[1] in Fig. 3 shows the structure of the data (the point types correspond to the cluster identities A, B, C, and D).

2.2 Constraints and background distribution

The user interaction consists of selecting a point set (which we refer to as a *cluster*), studying the statistics of this cluster, and possible *marking* this cluster. Subsequently, the system provides a new visualization showing structure complementary to the structure encoded in the background distribution. To implement the envisioned interaction scheme, we wish to define constraints (specifications of the data) and to construct a background distribution such that the constraints set by the user are satisfied. Intuitively, the more constraints we have, the closer the distribution should be to the true data, since the constraints added are based on the data. Typically, the constraints will not be sufficient to define a distribution, because there are still many degrees of freedom. Arguably, the most neutral distribution is the distribution of maximum entropy (MaxEnt), because that is the only distribution which does not add any side information (Cover and Thomas 2005).

We must also define *some* initial background distribution. A reasonable and convenient assumption is that the initial background distribution equals a spherical Gaussian distribution with zero mean and unit variance, given by

$$q(\mathbf{X}) \propto \exp\left(-\sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{x}_i / 2\right). \tag{1}$$

This is equivalent to the MaxEnt distribution with known mean and variance (but not co-variance) for all attributes. If we normalize the data, these statistics are obviously zero and one respectively, for every attribute.

As illustrated in Fig. 1, the interaction process is such that the user is shown 2-D projections (Fig. 1c) where the data and the background distribution differ the most. The initial view shown to the user is a projection of the whitened data (see Sect. 2.5) onto the first two PCA components or the two ICA components with the highest score, whichever of these projection methods the user deems more appropriate.

Example 2 Figure 4a shows the projection of the whitened $\hat{\mathbf{X}}_5$ onto the two ICA components with the highest scores using log-cosh objective function. One can observe the cluster structure in the first three dimensions X1–X3. The gray points represent a sample from the background distribution. When shown together with the data, it becomes evident that the data and the background distribution differ.

Subsequently, we can define constraints on subsets of points in $\mathbb{R}^{n \times d}$ for a given projection by introducing *linear and quadratic constraint functions* (Lijffijt et al. 2018). A constraint is parametrized by the subset of rows $I \subseteq [n]$ that are involved and a projection vector $\mathbf{w} \in \mathbb{R}^d$. The *linear constraint function* is defined by

[1] A plot with scatterplots for all the pairs of variables together with the distribution of each variable at the diagonal.

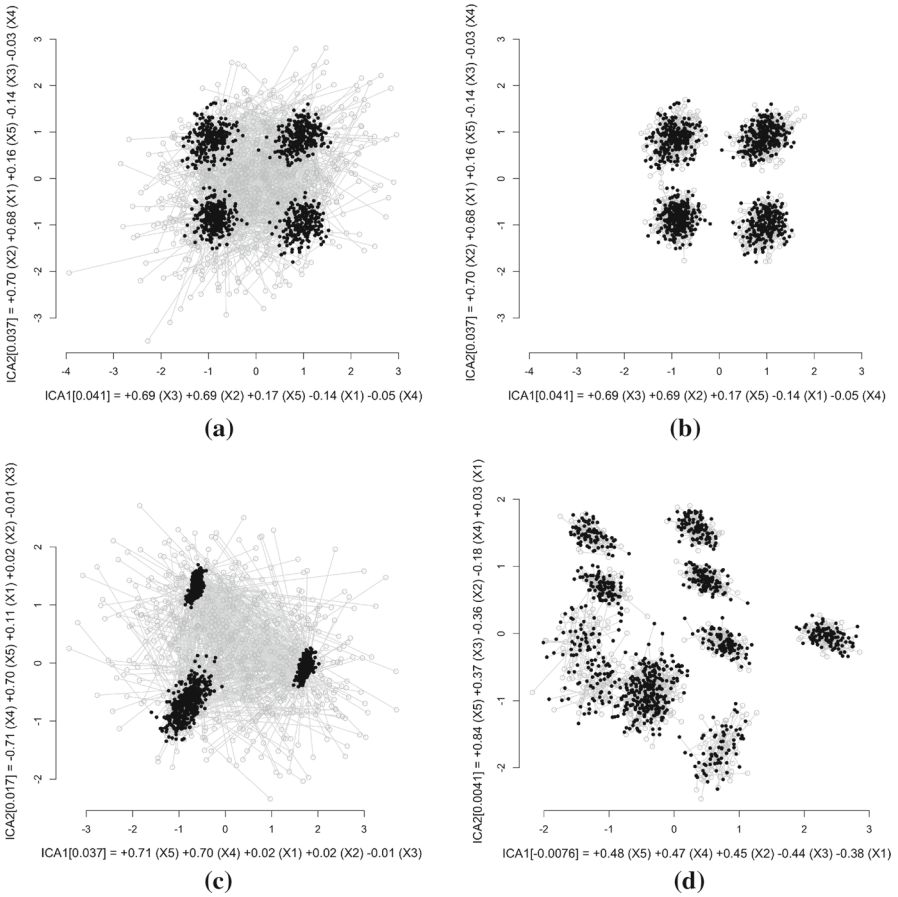

Fig. 4 **a** The whitened synthetic data $\hat{\mathbf{X}}_5$ projected into the two ICA components with the highest scores using log-cosh objective function and shown together with a sample of the background distribution (in gray). With no knowledge about the belief state of the user, the background distribution is Gaussian. **b** The same projection with the background distribution updated to take into account cluster constraints for the four visible clusters. **c** The next most informative ICA projection for $\hat{\mathbf{X}}_5$. **d** The ICA projection obtained after further cluster constraints for the three visible clusters in **c** have been added and the background distribution has been updated

$$f_{\text{lin}}(\mathbf{X}, I, \mathbf{w}) = \sum\nolimits_{i \in I} \mathbf{w}^T \mathbf{x}_i, \tag{2}$$

and the *quadratic constraint function* by

$$f_{\text{quad}}(\mathbf{X}, I, \mathbf{w}) = \sum\nolimits_{i \in I} \left(\mathbf{w}^T \left(\mathbf{x}_i - \hat{\mathbf{m}}_I \right) \right)^2, \tag{3}$$

where we have used

$$\hat{\mathbf{m}}_I = \sum\nolimits_{i \in I} \hat{\mathbf{x}}_i / |I|. \tag{4}$$

These constraint functions specify the mean and variance for a set of points, for a specific direction $\mathbf{w}$. Notice that $\hat{\mathbf{m}}_I$ is not a random variable but a constant that depends on the observed data. If it were a random variable, it would introduce cross-terms between rows and the distribution would no longer be independent for different rows. In principle, we could set $\hat{\mathbf{m}}$ to any constant value, including zero. However, for the numerical algorithm to converge quickly, we use the value specified by Eq. (4).

We denote a constraint by a triplet $C = (c, I, \mathbf{w})$, where $c \in \{lin, quad\}$, and the constraint function is then given by $f_c(\mathbf{X}, I, \mathbf{w})$. We can now use the linear and quadratic constraint functions to express several types of knowledge a user may have about the data, e.g., knowledge of a cluster in the data or the marginal distribution of the data, which we can then encode into the background distribution.

To start with, we can encode the mean and variance, i.e., the first and second moment of the marginal distribution, of each attribute:

(i) *Margin constraint* consists of a linear and a quadratic constraint for each of the columns in $[d]$, respectively, the total number of constraints being $2d$.

We can encode the mean and (co)variance statistics of a point cluster for all attributes:

(ii) *Cluster constraint* is defined as follows. We make a singular value decomposition (SVD) of the points in the cluster defined by I. Then a linear and a quadratic constraint is defined for each of the eigenvectors. This results in $2d$ constraints per cluster.

We can also encode the mean and (co)variance statistics of the full data for all attributes:

(iii) *1-cluster constraint* is a special case of a cluster constraint where the full dataset is assumed to be in one single cluster (i.e., $I = [n]$). Essentially, this means that the data is modeled by its principal components and the correlations are taken into account, unlike with the marginal constraints, again resulting to $2d$ constraints.

Finally, we can encode the mean and variance of a point cluster or the full data as shown in the current 2-D projection:

(iv) *2-D constraint* consists of a linear and a quadratic constraint for the two eigen vectors spanning the 2-D projection in question, resulting to 4 constraints.

2.3 Updating the background distribution

Having formalized the constraints, we are now ready to formulate our main problem, i.e., how to update the background distribution given a set of constraints.

Problem 1 Given a dataset $\hat{\mathbf{X}}$ and k constraints $\mathscr{C} = \{C^1, \ldots, C^k\}$, find a probability density p over datasets $\mathbf{X} \in \mathbb{R}^{n \times d}$ such that the entropy defined by

$$S = -E_{p(\mathbf{X})}\left[\log\left(p(\mathbf{X})/q(\mathbf{X})\right)\right] \tag{5}$$

is maximized, while the following constraints are satisfied for all $t \in [k]$:

$$E_{p(\mathbf{X})}\left[f_{c^t}(\mathbf{X}, I^t, \mathbf{w}^t)\right] = \hat{v}^t, \tag{6}$$

where $\hat{v}^t = f_{c^t}(\hat{\mathbf{X}}, I^t, \mathbf{w}^t)$ and $q(\mathbf{X}) \propto \exp\left(-\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i/2\right)$.

The distribution p that is a solution to the Problem 1 is the background distribution taking into account $\mathscr{C}$. Intuitively, the background distribution is the maximally random distribution such that the constraints are preserved in expectation. Due to our choice of the initial background distribution and the constraint functions, the MaxEnt solution to Problem 1 is a multivariate Gaussian distribution. The form of the solution to Problem 1 is given by the following lemma.

Lemma 1 *The probability density p that is a solution to Problem 1 is of the form*

$$p(\mathbf{X}) \propto q(\mathbf{X}) \times \exp\left(\sum_{t=1}^k \lambda^t f_{c^t}(\mathbf{X}, I^t, \mathbf{w}^t)\right), \tag{7}$$

where $\lambda^t \in \mathbb{R}$ are real-valued parameters.

See, e.g., Cover and Thomas (2005, Chapter 12) for a proof.

We make an observation that adding a margin constraint or 1-cluster constraint to the background distribution is equivalent to transforming the data to zero mean and unit variance or whitening of the data, respectively.

Equation (7) can also be written in the form

$$p(\mathbf{X} \mid \theta) \propto \exp\left(-\sum_{i=1}^n (\mathbf{x}_i - \mathbf{m}_i)^T \Sigma_i^{-1} (\mathbf{x}_i - \mathbf{m}_i)/2\right), \tag{8}$$

using the natural parameters collectively denoted by

$$\theta = \{\theta_i\}_{i\in[n]} = \left\{\left(\Sigma_i^{-1}\mathbf{m}_i, \Sigma_i^{-1}\right)\right\}_{i\in[n]}.$$

By matching the terms linear and quadratic in $\mathbf{x}_i$ in Eqs. (7) and (8), we can write Eq. (8) as sums of the terms of the form $\lambda^t f_{c^t}(\mathbf{X}, I^t, \mathbf{w}^t)$. The dual parameters are given by $\mu = \{\mu_i\}_{i\in[n]} = \{(\mathbf{m}_i, \Sigma_i)\}_{i\in[n]}$ and can be obtained from the natural parameters by using matrix inversion and multiplication operations.

Problem 1 can be solved numerically as follows. Initially, we set the lambda parameters to $\lambda^1 = \cdots = \lambda^k = 0$, with the natural dual parameters then given by $\theta_i = \mu_i = (\mathbf{0}, \mathbf{1})$ for all $i \in [n]$. Given a set of constraints, the lambda parameters are updated iteratively as follows. Given some values for the lambda parameters and the respective natural and dual parameters, we choose a constraint $t \in [k]$ and find a value for λ^t such that the constraint in Eq. (6) is satisfied for this chosen t. We then iterate this process for all constraints $t \in [k]$ until convergence. Due to the convexity of the problem, we are always guaranteed to eventually end up in a globally optimal solution. For a given set of lambda parameters, we can then find the natural parameters in θ by simple addition, and the dual parameters μ using θ. Finally, the

expectation in Eq. (6) can be computed by using the dual parameters and the identities $E_{p(\mathbf{X}|\theta)}\left[\mathbf{x_i}\mathbf{x_i}^T\right] = \Sigma_i + \mathbf{m}_i\mathbf{m}_i^T$ and $E_{p(\mathbf{X}|\theta)}\left[\mathbf{x}_i\right] = \mathbf{m}_i$.

Example 3 After observing the view in Fig. 4a the user can add a cluster constraint for each of the four clusters visible in the view. The background distribution is then updated to take into account the added constraints by solving Problem 1. In Fig. 4b a sample of the updated background distribution (gray points) is shown together with the data (black points).

2.3.1 Update rules

A straightforward implementation of the above-mentioned optimization process is inefficient because we need to store parameters for n rows and the matrix inversion is an $O(d^3)$ operation, resulting to a time complexity of $O(nd^3)$. We can, however, substantially speed up the computations using two observations. First, two rows affected by the exactly same set of constraints will have equal parameters, i.e., we have $\theta_i = \theta_j$ and $\mu_i = \mu_j$ for such rows i and j. Thus, we need only to store and compute values of the parameters θ_i and μ_i for "equivalence classes" of rows, whose number depends on the number and the overlap of the constraints, but not on n. Second, if we store both the natural and dual parameters at each iteration, the update due to each constraint corresponds to a rank-1 update to the covariance matrix Σ_i^{-1}. We can then use the Woodbury Matrix Identity taking $O(d^2)$ time to compute the inverse, instead of $O(d^3)$.

A further observation is that by storing the natural and dual parameters at each step, we do not need to explicitly store the values of the lambda parameters. At each iteration we are only interested in the change of λ^t instead of its absolute value. After these speedups, we expect the optimization process to take $O(d^2)$ time per constraint and to be asymptotically independent of n. For simplicity, in the following description, we retain the sums of the form $\sum_{i \in I^t}$. However, in the implementation we replace these by the more efficient weighted sums over the equivalence classes of rows. To simplify and clarify the notation we use parameters with a tilde (e.g., $\tilde{\Sigma}$) to denote them before the update and parameters without (e.g., Σ) to denote the values after the update, and λ to denote the change in λ^t.

For a *linear constraint* t the expectation is given by

$$v^t = E_{p(\mathbf{X}|\theta)}\left[f_{lin}(\mathbf{X}, I^t, \mathbf{w}^t)\right] = \sum\nolimits_{i \in I^t} \mathbf{w}^{tT}\mathbf{m}_i.$$

The update rules for the parameters are given by $\theta_{i1} = \tilde{\theta}_{i1} + \lambda\mathbf{w}^t$ and $\mu_{i1} = \Sigma_i\theta_{i1}$. Solving for $v^t = \hat{v}^t$ gives the required change in λ^t as

$$\lambda = \left(\hat{v}^t - \tilde{v}^t\right) / \left(\sum\nolimits_{i \in I^t} \mathbf{w}^{tT}\tilde{\Sigma}_i\mathbf{w}^t\right), \tag{9}$$

where $\tilde{v}^t$ denotes the value of v^t before the update. Notice the change in λ^t is zero if $\tilde{v}^t = \hat{v}^t$, as expected.

For a *quadratic constraint t* the expectation is given by

$$v^t = E_{p(\mathbf{X}|\theta)}\left[f_{quad}(\mathbf{X}, I^t, \mathbf{w}^t)\right] = \mathbf{w}^{tT} \sum\nolimits_{i \in I^t} \left(\Sigma_i + \mathbf{q}_i \mathbf{q}_i^T\right) \mathbf{w}^t,$$

where $\mathbf{q}_i = \mathbf{m}_i - \hat{\mathbf{m}}_{I^t}$. The update rules for the parameters are

$$\begin{aligned}
\theta_{i1} &= \tilde{\theta}_{i1} + \lambda \delta \mathbf{w}^t, \\
\theta_{i2} &= \tilde{\theta}_{i2} + \lambda \mathbf{w}^t \mathbf{w}^{tT}, \\
\mu_{i1} &= \Sigma_i \theta_{i1}, \text{ and} \\
\mu_{i2} &= \tilde{\Sigma}_i - \lambda \mathbf{g}_i \mathbf{g}_i^T / \left(1 + \lambda \mathbf{w}^{tT} \mathbf{g}_i\right),
\end{aligned}$$

where have used the short-hands $\delta = \hat{\mathbf{m}}_{I^t}^T \mathbf{w}^t$ and $\mathbf{g}_i = \tilde{\Sigma}_i \mathbf{w}^t$. We use the Woodbury Matrix Identity to avoid explicit matrix inversion in the computation of μ_{i2}. Again, solving for $v^t = \hat{v}^t$ gives an equation

$$\phi(\lambda) = \sum\nolimits_{i \in I^t} \left(\Lambda_i c_i^2 - f_i^2 c_i^2 + 2 f_i c_i (\delta - e_i)\right) + \hat{v}^t - \tilde{v}^t = 0, \tag{10}$$

where we have used the following shorthands

$$\begin{aligned}
\mathbf{b}_i &= \tilde{\Sigma}_i \tilde{\theta}_{i1}, \\
c_i &= \mathbf{b}_i^T \mathbf{w}^t, \\
\Lambda_i &= \lambda / (1 + \lambda c_i), \\
d_i &= \mathbf{b}_i^T \tilde{\theta}_{i1}, \\
e_i &= \tilde{\mathbf{m}}_i^T \mathbf{w}^t, \text{ and} \\
f_i &= \lambda \delta - \Lambda_i d_i - \Lambda_i \lambda \delta c_i.
\end{aligned}$$

Notice that Λ_i and f_i are functions of λ. We conclude with the observation that $\phi(\lambda)$ is a monotone function, whose root can be determined efficiently with a one-dimensional root-finding algorithm.

2.4 About convergence

In the runtime experiment (Sect. 3.1, Table 2) we define the optimization to be converged when the maximal absolute change in the lambda parameters is 10^{-2} or when the maximal change in the means or square roots of variance constraints is at most 10^{-2} times the standard deviation of the full data. We describe in this section a situation where the convergence is very slow, and a fixed time cutoff becomes useful. The iteration is guaranteed to converge eventually, but in certain cases—especially if the size of the dataset (n) is small or the size of some clusters ($|I^t|$) is small compared to the dimensionality of the dataset (d)—the convergence can be slow, as shown in the following adversarial example.

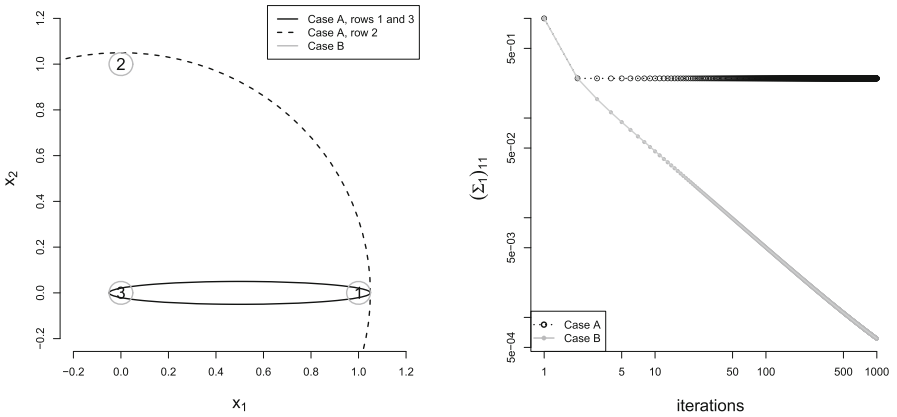

Fig. 5 **a** Adversarial toy data and 1σ confidence ellipsoids (expanded a bit for visual clarity in zero variance directions) for the background distribution. **b** Convergence of $(\Sigma_1)_{11}$ for two sets of constraints $\mathscr{C}_A$ (black line) and $\mathscr{C}_B$ (gray line)

Example 4 Consider a dataset of three points ($n = 3$) in two dimensions ($d = 2$), shown in Fig. 5a and given by

$$\hat{\mathbf{X}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \tag{11}$$

and two sets of constraints:

(A) The first set of constraints consists of the cluster constraints related to the first and the third row and is given by $\mathscr{C}_A = \{C^1, \ldots, C^4\}$, where $c^1 = c^3 = lin$, $c^2 = c^4 = quad$, $I^1 = \cdots = I^4 = \{1, 3\}$, $w^1 = w^2 = (1, 0)^T$, and $w^3 = w^4 = (0, 1)^T$.

(B) The second set of constraints has an additional cluster constraint related to the second and the third row and is given by $\mathscr{C}_B = \{C^1, \ldots, C^8\}$, where $C^1, \ldots, C^4$ are as above and $c^5 = c^7 = lin$, $c^6 = c^8 = quad$, $I^5 = \cdots = I^8 = \{2, 3\}$, $w^5 = w^6 = (1, 0)^T$, and $w^7 = w^8 = (0, 1)^T$.

Next, we consider convergence when solving Problem 1 using these two sets of constraints.

Case A The solution to Problem 1 with constraints in $\mathscr{C}_A$ is given by $\mathbf{m}_1 = \mathbf{m}_3 = (\frac{1}{2}, 0)^T$, $\mathbf{m}_2 = (0, 0)^T$,

$$\Sigma_1 = \Sigma_3 = \begin{pmatrix} \frac{1}{4} & 0 \\ 0 & 0 \end{pmatrix}, \quad \text{and} \quad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{12}$$

Note that if the number of data points in a cluster constraint is at most the number of dimensions in the data, there are necessarily directions in which the variance of the

background distribution is zero, see Fig. 5a. However, since we have here a single cluster constraint with no overlapping constraints, the convergence is very fast and, in fact, occurs after one pass over the lambda variables as shown in Fig. 5b (black line).

Case B The solution to Problem 1 with constraints in $\mathscr{C}_B$ are given by $\mathbf{m}_1 = (1, 0)^T$, $\mathbf{m}_2 = (0, 1)^T$, $\mathbf{m}_3 = (0, 0)^T$, and

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{13}$$

Here we observe that adding a second overlapping cluster constraint, combined with the small variance directions in both of the constraints restricts the variance of the third data point to zero. Because both of the clusters have only one additional data point, it follows that the variance of all data points is then zero. The small variance and the overlapping constraints for data points cause the convergence here to be substantially slower, as shown in Fig. 5b (gray line). The variance scales roughly as $(\Sigma_1)_{11} \propto \tau^{-1}$, where τ is the number of optimization steps, the global optimum being in singular point at $(\Sigma_1)_{11} = 0$.

The slow convergence in the above example is due to the overlapping of constraints and the quadratic constraints with a small variance (caused here by the small number of points per cluster). A way to speed up the convergence would be—perhaps unintuitively—to add more data points: e.g., to replicate each data point 10 times with random noise added to each replicate. When a data point would be selected to a constraint, then all of its replicates would be included as well. This would set a lower limit on the variance of the background model and hence, would be expected to speed up the convergence. Another way to solve the issue is just to cut off the iterations after some time point leading up to a larger variance than in the optimal solution. The latter approach appears to be typically acceptable in practice.

2.5 Whitening operation for finding the most informative visualization

Once we have found the distribution that solves Problem 1, the next task is to find and visualize the maximal differences between the data and the background distribution defined by Eq. (5).

Here we use a *whitening operation* which is similar to ZCA-Mahalanobis whitening (Kessy et al. 2018) to find the directions in which the current background distribution p and the data differ the most. The underlying idea is that a direction-preserving whitening transformation of the data with p results in a unit Gaussian spherical distribution, if the data follows the current background distribution p. Thus, any deviation from the unit sphere distribution in the data whitened using p is a signal of difference between the data and the current background distribution.

More specifically, let the distribution p solving Problem 1 be parametrized by $\mu = \{(\mathbf{m}_i, \Sigma_i)\}_{i\in[n]}$ and consider $\mathbf{X} = (\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_n)^T \in \mathbb{R}^{n\times d}$. We define new whitened data vectors $\mathbf{y}_i$ as follows,

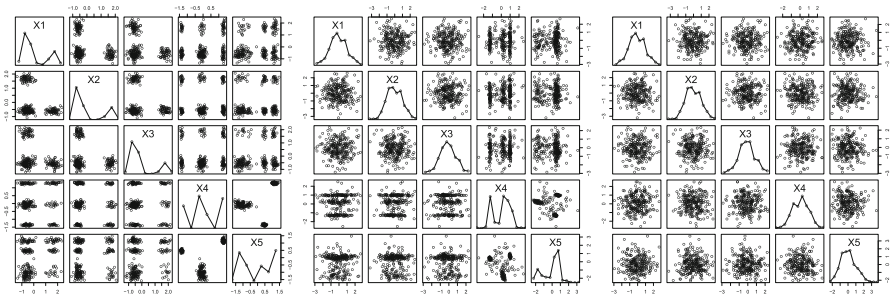

Fig. 6 A pairplot of the whitened data $\hat{\mathbf{Y}}_5$. **a** Initially, i.e., without constraints, $\hat{\mathbf{Y}}_5 = \hat{\mathbf{X}}_5$. **b** After the cluster constraints are added for the four clusters visible in Fig. 4a. **c** After further cluster constraints are added for the three clusters visible in Fig. 4c

$$\mathbf{y}_i = \Sigma_i^{-1/2} (\mathbf{x}_i - \mathbf{m}_i) , \tag{14}$$

where $\Sigma_i^{-1/2} = U_i D_i^{1/2} U_i^T$ with the SVD decomposition of Σ_i^{-1} given by $\Sigma_i^{-1} = U_i D_i U_i^T$, where U_i is an orthogonal matrix and D_i is a diagonal matrix. Notice that if we used one transformation matrix for the whole data, this would correspond to the normal whitening transformation (Kessy et al. 2018). However, here we may have a different transformation for each of the rows. Furthermore, normally the transformation matrix would be computed from the data, but here we compute it from the constrained model, i.e., using the background distribution.

It is easy to see that if $\mathbf{x}_i$ obeys the distribution of Eq. (8), then $D_i^{1/2} U_i^T (\mathbf{x}_i - \mathbf{m}_i)$ obeys unit spherical distribution. Hence, any rotation of this vector obeys a unit sphere distribution as well. We rotate this vector back to the direction of $\mathbf{x}_i$ so that after the final rotation, the vectors $\mathbf{y}_i$ for different rows i have a comparable direction.

Now, we apply the whitening transformation on our data matrix $\hat{\mathbf{X}}$ and use $\hat{\mathbf{Y}} = \left(\hat{\mathbf{y}}_1 \hat{\mathbf{y}}_2 \ldots \hat{\mathbf{y}}_n\right)^T$ to denote the whitened data matrix. Notice that when there are no constraints, that is $\mathbf{m}_i = \mathbf{0}$ and $\Sigma_i^{-1} = \mathbf{1}$, the whitening operation reduces to identity operation, i.e., $\hat{\mathbf{Y}} = \hat{\mathbf{X}}$.

Example 5 To illustrate the whitening operation, we show in Fig. 6 pairplots of the whitened data matrix $\hat{\mathbf{Y}}_5$ for the synthetic data $\hat{\mathbf{X}}_5$ and different background distributions (i.e., sets of constraints). Initially, i.e., without any constraints (Fig. 6a) the whitened data matches $\hat{\mathbf{X}}_5$. Figure 6b shows the whitened data after the background distribution has been updated to take into account the addition of a cluster constraint for each of the four clusters in Fig. 4a. Now, in the first three dimensions X1–X3 the whitened data does not anymore significantly differ from Gaussian distribution, while in dimensions X4 and X5 it does.

In order to find directions where the data and the background distribution differ, i.e., the whitened data $\hat{\mathbf{Y}}$ differs from the unit Gaussian distribution with zero mean, an obvious choice is to use Principal Component Analysis (PCA) and look for directions

Table 1 ICA scores (sorted by absolute value) for all five components computed by FastICA for each of the iterative steps in Fig. 4

Projection	ICA scores				
Figure 4a, b	0.041	0.037	0.035	0.034	−0.015
Figure 4c	0.037	0.017	0.004	−0.003	−0.002
Figure 4d	−0.008	0.004	−0.003	0.003	−0.002

It can be seen that after the first set of constraints there are only two significantly non-Gaussian components left (both loadings on the fourth and fifth attribute of the data), while after the second set of constraints there is no substantial structure left in the data

in which the variance of $\hat{\mathbf{Y}}$ differs most from unity.[2] However, it may happen that the variance is already taken into account in the variance constraints. In this case, PCA becomes non-informative because all directions in $\hat{\mathbf{Y}}$ have equal mean and variance. Instead, we can use, e.g., Independent Component Analysis (ICA) and the FastICA algorithm (Hyvärinen 1999) with log-cosh G function as a default method to find non-Gaussian directions. To find the best two ICA components, we compute a full set of d components, and then take the two components that score best on the log-cosh objective function. Clearly, when there are no constraints, our approach equals standard PCA and ICA on the original data, but when there are constraints, the output will be different.

To be able to visualize the background distribution together with the data in the found projection, we use a random dataset that can be obtained by sampling a data point for each $i \in [n]$ from the multivariate Gaussian distribution parametrized by θ_i.

Example 6 The directions in which the whitened data $\hat{\mathbf{Y}}_5$ in Fig. 6b differs the most from Gaussian (using ICA) are shown in Fig. 4c. The user can observe the cluster structure in dimensions X4 and X5, which would not be possible to find with non-iterative methods. Furthermore, it is clear that the sample from the background distribution (the points shown in gray in Fig. 4) is different from the data in this projection. After adding a cluster constraint for each of the three visible clusters, the updated background distribution becomes a faithful representation of the data, and thus the whitened data shown in Fig. 6c resembles a unit Gaussian spherical distribution in all dimensions. This is also reflected in a visible drop in ICA scores in Table 1.

2.6 A summary of the proposed interactive framework for EDA

Now, we are ready to summarize our framework. Initially, we have the dataset $\hat{\mathbf{X}}$, the set of constraints $\mathscr{C}$ is empty, and the background distribution equals a spherical Gaussian distribution with zero mean and unit variance (Eq. 1). At each iteration, the following steps are performed, and the exploration continues as long as the user is convinced that she has observed relevant features of the data (i.e., there is now visible difference between the background distribution and the data).

[2] Here we measure the difference of variance from unity to the direction of each principal component by $(\sigma^2 - \log \sigma^2 - 1)/2$, where σ^2 is the variance to the direction of the principal component, and show in the scatterplot the two principal components with the largest differences from unity.

1. The data $\hat{\mathbf{X}}$ is whitened with respect to the background distribution (Eq. 14).
2. The first two PCA or ICA components of the whitened data $\hat{\mathbf{Y}}$ are computed to obtain the *most informative projection* with respect to the current knowledge.
3. The data $\hat{\mathbf{X}}$ and a sample from the background distribution are projected into the directions found in Step 2.
4. In the projection, the user may observe differences between the data and the background distribution. She then formulates the observations in terms of constraints $\{C_1, \ldots, C_k\}$, and the set of constraints is updated to $\mathscr{C} = \mathscr{C} \cup \{C_1, \ldots, C_k\}$.
5. The background distribution is updated to take into account the added constraints, i.e., Problem 1 is solved with respect to the updated $\mathscr{C}$.
6. The process continues from Step 1.

Remark 1 If the user has prior knowledge about the data, this can represented using a set of constraints $\mathscr{C} \neq \emptyset$. Then, one should use the distribution p that is a solution to Problem 1 with respect to $\mathscr{C}$ as the initial background distribution instead of using a spherical Gaussian distribution with zero mean and unit variance.

Remark 2 Throughout the process the background distribution has the form of a multivariate Gaussian distribution with mean and co-variance that may differ from point to point. This is not by assumption, but it is the result of the MaxEnt principle along with constraints that specify the mean and variance, which leads to a Gaussian distribution.

3 Experiments

In this section, we demonstrate the use of our framework in exploratory data analysis. The implementation of steps needed for the exploration flow described in Sect. 2.6 is done using R 3.4.0 (R Core Team 2017) and FASTICA (Marchini et al. 2013). In addition, we have implemented an interactive proof-of-concept system SIDER (Puolamäki 2019), see Sect. 3.4, using SHINY (Chang et al. 2017) for the user interface. The code implementing our framework and the SIDER system together with the code to run the experiments (Sects. 2.4, 3) has been released as a free open source software under the MIT license at https://github.com/edahelsinki/sideR (Last Accessed: 28 Aug 2019).

Our focus here is to show how our approach is able to provide the user with insightful projections of data and reveal the differences between the background distribution and the data. We start by a runtime experiment in which we test the model with data set sizes typical for interactive systems and visual exploration, i.e., there are on the order of thousands of data points. *If there are more data points, it often makes sense to downsample the data first*. Following the runtime experiment, we use real datasets to illustrate how we are able to find relevant projections for the user and display differences between the background distribution and the data.

3.1 Runtime experiment

We generated synthetic datasets parametrized by the number of data points (n), the dimensionality of the data (d), and the number of clusters (k). Each dataset was created

Table 2 Median wall clock running times, based on 10 runs for each set of parameters for finding the correct parameters (OPTIM) and running the ICA (ICA) algorithm without time cutoff

n	d	OPTIM	ICA
2048	16	{0.0, 0.2, 0.3, 0.5}	{0.6, 0.6, 0.6, 0.6}
2048	32	{0.0, 0.6, 1.0, 2.1}	{1.5, 1.5, 1.6, 1.6}
2048	64	{0.1, 2.7, 5.2, 11.0}	{5.1, 5.2, 4.9, 4.9}
2048	128	{1.2, 21.4, 48.1, 124.6}	{17.8, 17.6, 17.4, 17.0}
4096	16	{0.0, 0.2, 0.3, 0.5}	{1.1, 1.1, 1.1, 1.1}
4096	32	{0.0, 0.6, 1.0, 2.0}	{3.1, 3.4, 3.0, 3.1}
4096	64	{0.2, 2.5, 6.0, 11.6}	{9.8, 9.3, 9.5, 9.6}
4096	128	{1.2, 23.4, 56.4, 121.3}	{34.2, 34.7, 34.4, 34.4}
8192	16	{0.0, 0.2, 0.3, 0.6}	{2.6, 2.2, 2.5, 2.1}
8192	32	{0.0, 0.6, 1.0, 2.0}	{6.5, 6.0, 5.9, 5.9}
8192	64	{0.2, 2.7, 6.0, 12.2}	{20.7, 20.4, 19.8, 20.1}
8192	128	{1.2, 21.9, 44.1, 110.3}	{67.9, 67.5, 67.1, 67.6}

The columns OPTIM and ICA list the running times in seconds for $k \in \{1, 2, 4, 8\}$

by first randomly sampling k cluster centroids and then allocating data points around each of the centroids. We added column constraints ($2d$ constraints) for each dataset and for the datasets with $k > 1$ we additionally used cluster constraints for each of the k clusters in the data ($2dk$ constraints). In Table 2 the median wall clock running times are provided without any cut-off, based on 10 runs for each set of parameters, ran on a Apple MacBook Air with 2.2 GHz Intel Core i7 processor and a single-threaded R 3.4.0 implementation of the algorithm.

The algorithm is first initialized (INIT) which is typically very fast, after which the correct parameters are found (OPTIM). Then preprocessing is done for sampling and whitening (PREPROCESS) after which we produce a whitened dataset (WHITENING) and a random sample of the MaxEnt distribution (SAMPLE). These are then used to run the PCA (PCA) and ICA (ICA) algorithms. We found that INIT, PREPROCESS, WHITENING, SAMPLE, and PCA always take less than 2 s each and they are not reported in the table. Most of the time is consumed by OPTIM. We observe in Table 2 that, as expected, the time consumed does not depend on the number of rows n in the dataset. Each of the optimization steps takes $O(d^2)$ time per constraint and there are $O(kd)$ constraints, hence the time consumed scales as expected roughly as $O(kd^3)$. In our following experiments with the real-world datasets, we stop the optimization after a time cut-off of 10 s, even when convergence has not been achieved. For larger matrices the time consumed by ICA becomes significant, scaling roughly as $O(nd^2)$.

3.2 British National Corpus data

The British National Corpus (BNC 2007) is one of the largest annotated text corpora freely available in full-text format. The texts are annotated with information such as author gender, age, and target audience, and all texts have been classified into genres

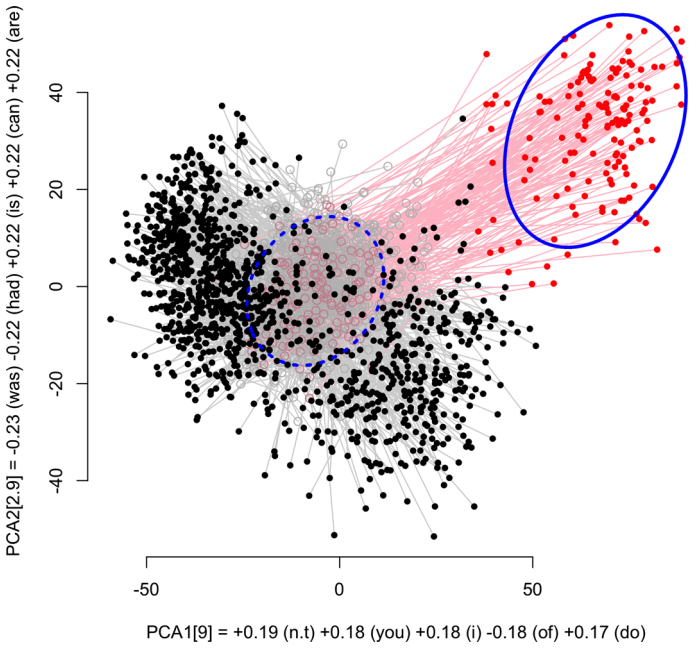

Fig. 7 A use case with the BNC data. The data (solid black/red points) projected into the first two PCA components for the whitened data (i.e., the most informative PCA projection with respect to the initial background distribution) shown together with a sample from the background distribution (gray/pink circles), with a gray/pink line connecting the data points with the corresponding sampled background points. The selection of points in red appears to form a cluster different from the rest of the data (Color figure online)

(Lee 2001). As a high dimensional use case, we explore the high-level structure of the corpus. For a preprocessing step, we compute the vector-space model (word counts) using the first 2000 words from each text belonging to one the four main genres in the corpus ('prose fiction', 'transcribed conversations', 'broadsheet newspaper', 'academic prose') as done in Lijffijt and Nevalainen (2017). After preprocessing, we have word counts for 1335 texts and we use the 100 words with the highest counts as the dimensions and the main genres as the class information.

The initially most informative PCA projection to the BNC data is shown in Fig. 7. Here, in addition to data, a sample from the background distribution is shown by gray circles, with a gray line connecting the data points with the corresponding sampled background points. One should notice that the gray points and lines only provide a proxy for the difference between the data and the background distribution, since in reality the background distribution has a specified mean and covariance structure for every point. Nonetheless, this should illustrate broadly the density structure of the background distribution for the current projection, which we think is helpful to understand why the current visualization may provide new insights about the differences of the background distribution and the dataset.

Now, in the upper right corner, there is a group of points (red selection) that appears to form a cluster. We further visualize the points in the sample from the background

Table 3 Distribution of the class labels (genres) for the selections in the use case with the exploration of the BNC data

Genre	Selection		
	Figure 7	Figure 8a	All
'prose fiction'	9	17	426
'transcribed conversations'	142	0	144
'broadsheet newspaper'	0	267	280
'academic prose'	0	484	485
All	151	768	1335

distribution corresponding to the data points in the red selection using pink color (both for the circles and the connecting lines). The selected 151 points are mainly texts from 'transcribed conversations' (Jaccard-index to class 0.94), see Table 3 for the detailed distribution of genres in the selection. We also show in Fig. 7 in blue the 95% confidence ellipsoids for the distribution of the selection (solid blue ellipsoid) and the respective background samples (dotted blue ellipsoid) to aid in figuring out if the location of the selected points in the current projection differs substantially from the expected location under the background distribution.[3]

Next, we added a cluster constraint for the selection of data points shown in red in Fig. 7. We updated the the background distribution and whitened the data with respect to the updated background distribution. Using the whitened data, we computed the first two PCA components and obtained the projection shown in Fig. 8a. Here we selected another set of points differing from the background distribution (the selection in red in Fig. 8a). This set of points mainly contains 'academic prose' and 'broadsheet newspaper' (Jaccard-indices 0.63 and 0.35). After adding a cluster constraint for this selection, we updated the background distribution, whitened the data, and computed the PCA projection for the whitened data, resulting in the projection shown in Fig. 8b. Now, there is no apparent difference to the background distribution (reflected indeed in low PCA scores), and we conclude that the identified 'prose fiction' class, together with the combined cluster of 'academic prose' and 'broadsheet newspaper' explain the data well with respect to variation in counts of the most frequent words. Notice that we did not provide the class labels in advance, they were only used retrospectively.

Observe also that class labels are not necessary for gaining insights during the exploration, although for brevity we base most our discussion and observations on them. The location of clustered sets of points in the projection together with the weight vectors for the projection axes directly provides further information. For example, an inspection of the selection shown in Fig. 8a (mainly consisting of texts from classes 'broadsheet newspaper' and 'academic prose') shows that this selection has low values along PCA1 axis, which implies that the selected points consist of texts with a high frequency of the word 'of' and low frequencies of "n't" (as in "don't") and 'I'.

[3] The 95% confidence regions are here a visual aid computed from the points shown in the projection. The confidence ellipsoid could also be computed from the background distribution directly, but it is a simplification as well since every data point may have unique mean and co-variance parameters.

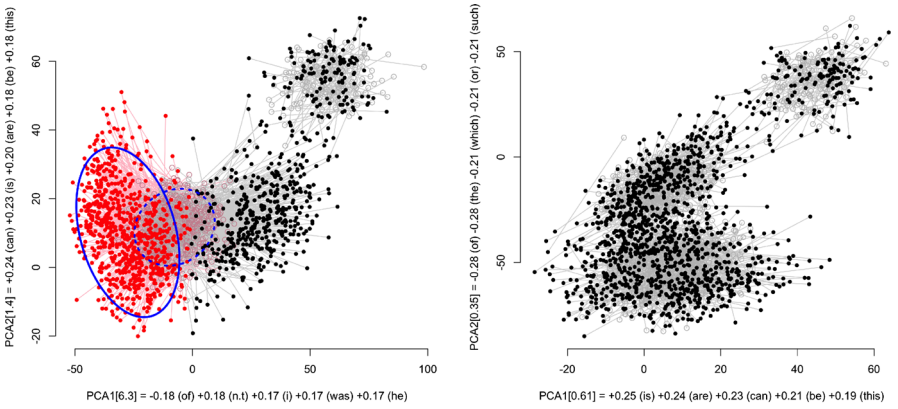

Fig. 8 The use case with the BNC data continues. **a** Selection of points for the second cluster constraint. The view is the most informative PCA projection with respect to the updated background distribution obtained after adding a cluster constraint for the points selected in Fig. 7. **b** After adding a further cluster constraint for the selection in red in **a** and updating of the background distribution, there is no longer a striking difference between the background distribution and the data in the most informative PCA projection (Color figure online)

3.3 UCI image segmentation data

As a second use case, we have the Image Segmentation dataset from the UCI machine learning repository (Dua and Graff 2019) with 2310 samples. The PCA projection (Fig. 9a) shows that the background distribution has a much larger variance than the data. Thus, we first added a 1-cluster constraint for the data (overall covariance) and updated the background distribution. After this, in the most informative projection (Fig. 9b) at least three sets of points quite clearly separated. The set of 330 points selected in Fig. 9b contains solely points from the class ‘sky’, while the 315 points in the lower left corner (selected in Fig. 9d) are from the class ‘grass’. The set of points clustered in the middle (selected in Fig. 9c) are mainly from the classes ‘brickface’, ‘cement’, ‘foliage’, ‘path’, and ‘window’ (with Jaccard-index approx. 0.2 each). The detailed distribution of class labels in the selections are provided in Table 4.

We next add a cluster constraint for each of the three selections, and show the data and a sample from the updated background distribution in Fig. 9e. We can observe that the background distribution now matches the data rather well with the exception of some outliers. Then, we whiten the data and compute the most informative PCA projection (Fig. 9f) which reveals that indeed there are outliers. For brevity, we did not continue the analysis, but the data obviously contains a lot more structure that we could explore in subsequent iterations. Furthermore, in many applications identifying and studying the outlier points deviating from the three-cluster main structure of the data could be interesting and useful.

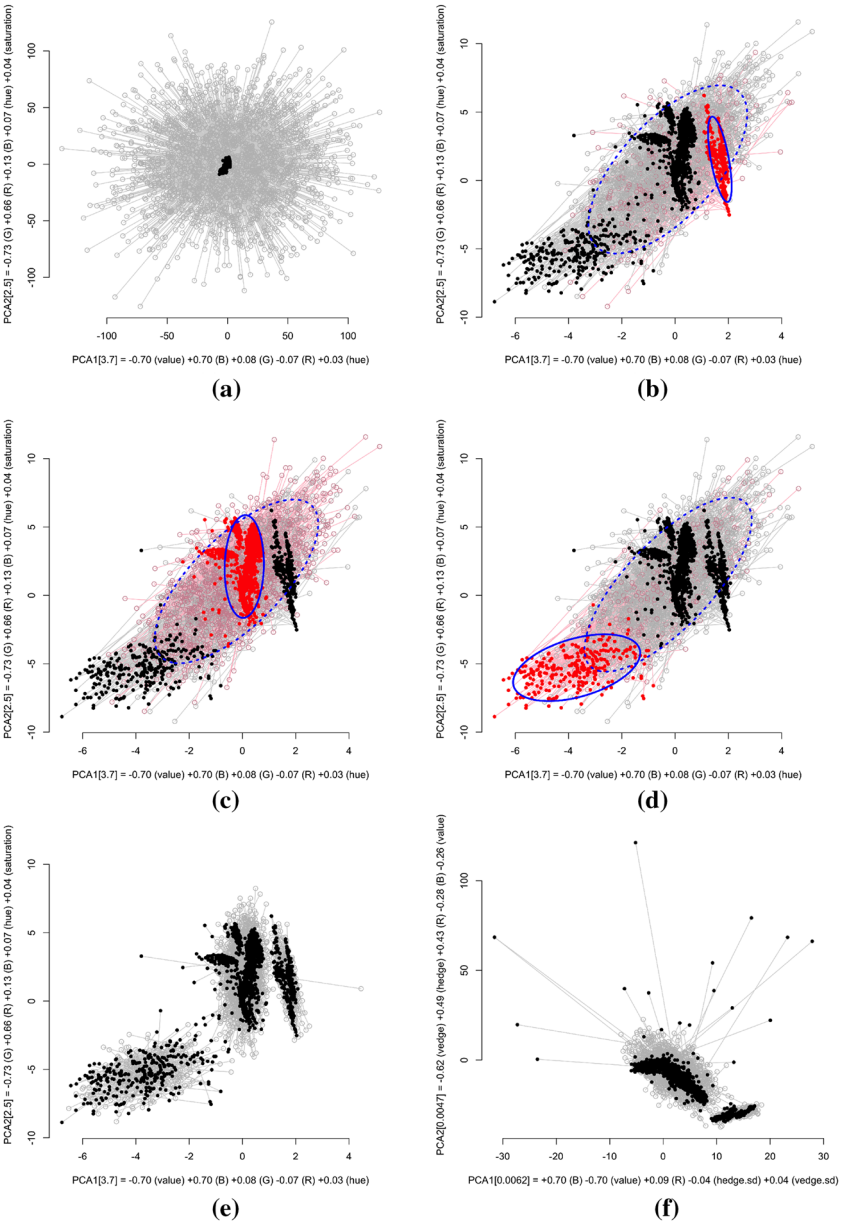

Fig. 9 A use case with the UCI image segmentation data. **a** Initially the scale of background distribution significantly differs from that of the data. **b** After adding a 1-cluster constraint and performing an update of the background distribution there is visible structure present in the most informative PCA projection. The points selected for the first cluster constraint are shown in red. **c**, **d** The selections of points for the second and third cluster constraint shown in red, respectively, in the same projection as **b**. **e** After these three cluster constraints are added and the background distribution is updated accordingly, the data and the background distribution are similar in this projection. **f** The most informative PCA projection for the data whitened using the updated background distribution shows mainly outliers (Color figure online)

Table 4 Distribution of the class labels for the selections in the use case with the exploration of the UCI Image Segmentation data

Class	Selection			
	Figure 9b	Figure 9c	Figure 9d	All
'brickface'	0	330	0	330
'cement'	0	330	0	330
'foliage'	0	330	0	330
'grass'	0	15	315	330
'path'	0	330	0	330
'sky'	330	0	0	330
'window'	0	329	0	330
All	330	1664	315	2310

3.4 Proof-of-concept system SIDER

We implemented an interactive demonstrator system SIDER (Puolamäki 2019) using our R implementation and SHINY (Chang et al. 2017). Our proof-of-concept system runs in the web browser using R as a back-end.

The user interface of SIDER is shown in Fig. 10. The main scatterplot (upper right corner) shows the PCA (here) or ICA projection of the data to directions in which the data and the background distribution differ the most. The tool uses the same visualization as used in the scatterplots in Sect. 3, i.e., the data is shown by solid black spheres and the currently selected subset of data points is shown by solid red spheres. A sample from the background distribution is shown by gray circles, with a gray line connecting the data points with the corresponding sampled background points. We further show in the main scatterplot in blue the 95% confidence ellipsoids for the distribution of the selection (solid blue ellipsoid) and the respective background samples (dotted blue ellipsoid).

We also show a pairplot (lower right corner) directly displaying the attributes maximally different with respect to the current selection (red points) as compared to the full dataset. In the left-hand panel, we show some statistics of the full data and of the data points that have been selected.

The user can add data points to a selection by directly marking them using a brushing operation, by using pre-defined classes that exist in the dataset, or by using previously saved groupings. The user can create a 2-D or cluster constraint of the current selection by clicking the appropriate button on the left-hand panel as well as recompute the background distribution to match the current constraints and update the projections. The user can also adjust convergence parameters which have by default been set so that the maximal time taken to update the background distribution is $\sim 10\,\mathrm{s}$ which is in practice typically more than enough. The interface has been designed so that time-consuming operations (taking more than $\sim 2\,\mathrm{s}$, i.e., updating the background distribution or computing the ICA projection) are executed only by a direct command by the user, which makes the system responsive and predictable.

The SIDER tool can be readily used to reproduce the exploration interactively for our use cases in Sects. 3.2 and 3.3. The visual user interface of SIDER makes selec-

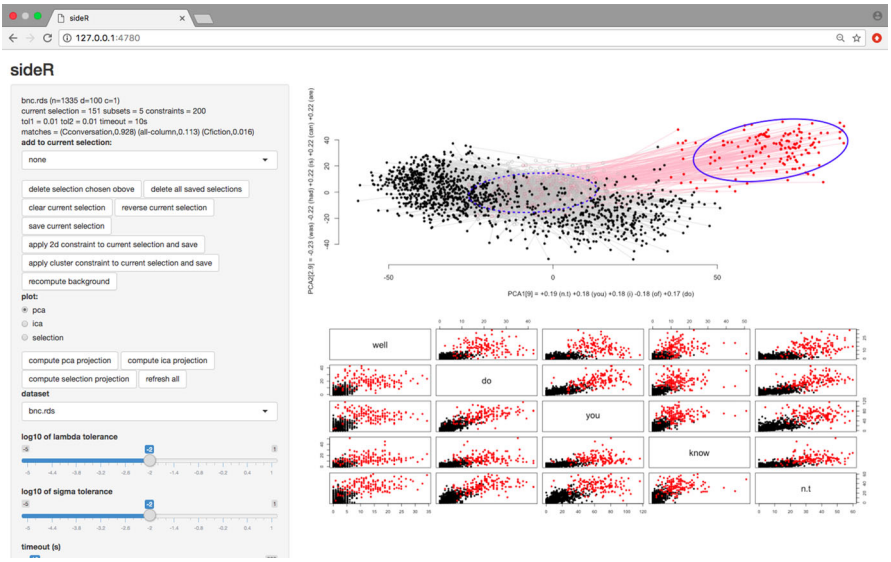

Fig. 10 The full user interface of SIDER. The data shown here is the British National Corpus data, see Sect. 3.2 for details

tion of sets of data points and the addition of new constraints easy. Furthermore, the implementation is fast enough to allow for comfortable interactive use for dataset sizes typical for visual exploration.

4 Related work

This work is motivated by the ideas in Puolamäki et al. (2016) and Kang et al. (2016) in which a similar system was constructed using constrained randomization. The constrained randomization approach (Hanhijärvi et al. 2009; Lijffijt et al. 2014) is similar to the MaxEnt distribution used here, but it relies on sampling of the data and no direct distribution assumptions are made. An advantage of the approach took here is that it is faster—which is essential in interactive applications—and scales more easily to larger data. Furthermore, here we have an explicit analytic form for the background distribution unlike in Puolamäki et al. (2016), where the background distribution was defined by a permutation operation.

The mathematical form of linear and quadratic constraints and efficient inference of the background distribution has been developed by us in Lijffijt et al. (2018). The presentation here is new and non-overlapping. The analytic form of the background distribution allows us, in addition to speeding up the computations, to define interestingness functions and the cluster constraint in a more natural manner. Here we also introduce the whitening method that allows our approach to be used with standard and robust projection pursuit methods such as PCA or ICA instead of the tailor-made line search algorithm of Puolamäki et al. (2016). Furthermore, we provide a fluent open source implementation written in R.

The Maximum Entropy method has been proposed as a part of Formalizing Subjective Interestingness (FORSIED) framework of data mining (De Bie 2011, 2013) modeling the user's knowledge by a background distribution. FORSIED has been studied in the context of dimensionality reduction and EDA (De Bie et al. 2016; Kang et al. 2018). Jaroszewicz and Simovici (2004) consider interestingness of frequent itemsets using Bayesian networks as background knowledge. Their approach is limited to categorical data and updating background knowledge requires modifying directly the Bayesian network. To the best of our knowledge, ours is the first instance in which this background distribution can be updated by direct interaction of the user, thus providing a principled method of EDA.

Several interactive dimensionality reduction methods have been proposed before, e.g., iPCA (Jeong et al. 2009), InVis (Paurat and Gärtner 2013), supervised PCA (Barshan et al. 2011), and guided locally linear embedding (Alipanahi and Ghodsi 2011). However, these methods aim to construct projections that *obey* the feedback, i.e., the projections are constrained by the feedback, in order to find a specific view of the data. The idea there is that the user and the system work together to construct a view. In contrast, in the proposed method the user feedback is used to track what the user has learned about the data in order to continuously provide new and complementary information about the data, working under the premise that a single view cannot capture all structure present in the data.

Many other special-purpose methods have been developed for active learning in diverse settings, e.g., in classification and ranking, as well as explicit models for user preferences. However, as these approaches are not targeted at data exploration, we do not review them here. Finally, several special-purpose methods have been developed for visual iterative data exploration in specific contexts, e.g., for itemset mining and subgroup discovery (Boley et al. 2013; Dzyuba and van Leeuwen 2013; van Leeuwen and Cardinaels 2015; Paurat et al. 2014), information retrieval (Ruotsalo et al. 2015), and network analysis (Chau et al. 2011).

The system presented here can be also considered to be an instance of *visually controllable data mining* (Puolamäki et al. 2010), where the objective is to implement advanced data analysis methods understandable and efficiently controllable by the user. Our approach satisfies the properties of a visually controllable data mining method, see (Puolamäki et al. 2010, Sect. 2.2): (VC1) the data and model space are presented visually, (VC2) there are intuitive visual interactions allowing the user to modify the model space, and (VC3) the method is fast enough for visual interaction.

Dimensionality reduction for EDA has been studied for decades starting with multidimensional scaling (MDS) (Kruskal 1964; Torgerson 1952) and Projection Pursuit (Friedman and Tukey 1974; Huber 1985). Recent research on this topic (referred to as manifold learning) is still inspired by MDS: find a low-dimensional embedding of points representing well the distances in the high-dimensional space. In contrast to PCA (Pearson 1901), the idea is to preserve small distances, and large distances are irrelevant, as long as they remain large, e.g., Locally Linear and (t-)Stochastic Neighbor Embedding (Hinton and Roweis 2003; Roweis and Saul 2000; van der Maaten and Hinton 2008). This is typically not possible to achieve perfectly, and a trade-off between precision and recall arises (Venna et al. 2010). Recent works are mostly spectral methods along this line.

In our framework, the whitening transformation allows us to reduce the problem of finding the most relevant direction in data with respect to the knowledge of the user into the problem of finding the direction in which the whitened data differs from the unit Gaussian distribution with zero mean. This allows us to use Projection Pursuit methods instead of having to define our own algorithm for the search. Thus, in the special case in which there is no prior knowledge about the data, our approach reduces to PCA/ICA.

5 Conclusions

There have been many efforts in the analysis of multivariate data in different contexts. For example, there are several Projection Pursuit and manifold learning methods using specific criteria to compress the data into a lower-dimensional—typically 2-D—presentation, while preserving features of interest. The inherent drawback of this approach is that the criteria for dimensionality reduction are defined typically in advance and it may or may not fit the user's need. It may be that a visualization shows only features of the data already known to the user, or features that are irrelevant for the task at hand.

The advantage of the dimensionality reduction methods is that the computer, unlike the human user, has a "view" of all the data and it can select a view in a more fine-tuned way and by using more complex criteria than a human could. A natural alternative to static visualizations using pre-defined criteria is the addition of interaction. The drawback of such interactions is, however, that they lack the sheer computational power utilized by the dimensionality reduction methods.

Our method fills the gap between automated dimensionality reduction methods and interactive systems. We propose to model the knowledge of a domain expert by a probability distribution computed by using the Maximum Entropy criterion. Furthermore, we propose powerful and yet intuitive interactions for the user to update the background distribution. Our approach uses Projection Pursuit methods and shows the directions in which the data and the background distribution differ the most. In this way, we utilize the power of Projection Pursuit at the same the allowing the user to adjust the criteria by which the computer chooses the directions to show her.

The current work presents a framework and a system for real-valued data and the background distribution which is modeled by multivariate Gaussian distributions. The same ideas could be generalized to other data types, such as categorical or ordinal data values, or to higher-order statistics, likely in a straightforward manner, as the mathematics of exponential family distribution would lead to similar derivations.

Our approach could also be extended to other interactions, especially in knowledge-intensive tasks. Instead of designing interactions directly and explicitly we can think that the "views of the data" (here mainly 2-D projections) and the interactions (here, e.g., marking the constraints) could also in other contexts be modeled as operations modifying the user's "background model". One of the main benefits of our approach is that the user marks the patterns she observes and thus the background distribution is always customized to the user's understanding of the data, without a need for assumptions such as that high variance directions are interesting to everyone, as implicitly

assumed when applying PCA for visualization. Furthermore, this approach cannot show user features that do not exist because the user is shown linear projections of the data.

The framework and the methodology proposed here are general and we anticipate our approach and tool to be useful in practice in the exploration of real-valued multidimensional datasets in various domain. As a concrete example, our approach and the SIDER tool appear promising for gating in computational flow cytometry. Gating is an analysis technique applied by biologists to flow cytometry data, where cells are data points and each point is described by a few intensity readings corresponding to emissions of different fluorescent dyes. The goal of gating is to extract clusters (gates) based on fluorescence intensities of the cells so that the cell types of a given sample can be differentiated. Initial experiments with up to tens of thousands of samples from flow-cytometry (Saeys et al. 2016) have shown the computations in SIDER to scale up well and the projections to reveal structure in the data potentially interesting to the application specialist.

Acknowledgements Open access funding provided by University of Helsinki including Helsinki University Central Hospital.

References

Alipanahi B, Ghodsi A (2011) Guided locally linear embedding. Pattern Recogn Lett 32(7):1029–1035. https://doi.org/10.1016/j.patrec.2011.02.002

Barshan E, Ghodsi A, Azimifar Z, Zolghadri Jahromi M (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recogn 44(7):1357–1371. https://doi.org/10.1016/j.patcog.2010.12.015

BNC (2007) The British National Corpus, v. 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. http://www.natcorp.ox.ac.uk/. Last Accessed 28 Aug 2019

Boley M, Mampaey M, Kang B, Tokmakov P, Wrobel S (2013) One click mining: interactive local pattern discovery through implicit preference and performance learning. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics, IDEA@KDD 2013, Chicago, IL, USA, August 11, 2013, pp 27–35. https://doi.org/10.1145/2501511.2501517

Chang W, Cheng J, Allaire J, Xie Y, McPherson J (2017) Shiny: web application framework for R. R package version 1.0.3. https://CRAN.R-project.org/package=shiny. Last Accessed: 28 Aug 2019

Chau D, Kittur A, Hong J, Faloutsos C (2011) Apolo: making sense of large network data by combining rich user interaction and machine learning. In: Proceedings of the international conference on human factors in computing systems, CHI 2011, Vancouver, BC, Canada, May 7–12, 2011, pp 167–176. https://doi.org/10.1145/1978942.1978967

Cover T, Thomas J (2005) Elements of information theory, 2nd edn. Wiley, Berlin

De Bie T (2011) An information-theoretic framework for data mining. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, CA, USA, August 21–24, 2011, pp 564–572. https://doi.org/10.1145/2020408.2020497

De Bie T (2013) Subjective interestingness in exploratory data mining. In: Advances in intelligent data analysis XII—12th international symposium, IDA 2013, London, UK, October 17–19, 2013 Proceedings, pp 19–31. https://doi.org/10.1007/978-3-642-41398-8_3

De Bie T, Lijffijt J, Santos-Rodriguez R, Kang B (2016) Informative data projections: a framework and two examples. In: 24th European symposium on artificial neural networks, ESANN 2016, Bruges, Belgium, April 27–29, 2016, pp 635–640

Dua D, Graff C (2019) UCI machine learning repository. http://archive.ics.uci.edu/ml. Last Accessed: 28 Aug 2019

Dzyuba V, van Leeuwen M (2013) Interactive discovery of interesting subgroup sets. In: Advances in intelligent data analysis XII—12th international symposium, IDA 2013, London, UK, October 17–19, 2013 Proceedings, pp 150–161. https://doi.org/10.1007/978-3-642-41398-8_14

Friedman J, Tukey J (1974) A projection pursuit algorithm for exploratory data analysis. IEEE Trans Comput 100(23):881–890. https://doi.org/10.1109/T-C.1974.224051

Hanhijärvi S, Ojala M, Vuokko N, Puolamäki K, Tatti N, Mannila H (2009) Tell me something I don't know: randomization strategies for iterative data mining. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France, June 28–July 1, 2009, pp 379–388. https://doi.org/10.1145/1557019.1557065

Hinton G, Roweis S (2003) Stochastic neighbor embedding. Adv Neural Inf Process Syst 15:857–864

Huber P (1985) Projection pursuit. Ann Stat 13(2):435–475

Hyvärinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans Neural Netw 10(3):626–634. https://doi.org/10.1109/72.761722

Jaroszewicz S, Simovici DA (2004) Interestingness of frequent itemsets using Bayesian networks as background knowledge. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, Seattle, Washington, USA, August 22–25, 2004, pp 178–186. https://doi.org/10.1145/1014052.1014074

Jeong DH, Ziemkiewicz C, Fisher B, Ribarsky W, Chang R (2009) iPCA: an interactive system for PCA-based visual analytics. Comput Graph Forum 28(3):767–774. https://doi.org/10.1111/j.1467-8659.2009.01475.x

Kang B, Puolamäki K, Lijffijt J, De Bie T (2016) A tool for subjective and interactive visual data exploration. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2016, Riva del Garda, Italy, September 19–23, 2016, Proceedings, Part III, pp 3–7. https://doi.org/10.1007/978-3-319-46131-1_1

Kang B, Lijffijt J, Santos-Rodríguez R, De Bie T (2018) SICA: subjectively interesting component analysis. Data Min Knowl Disc 32(4):949–987. https://doi.org/10.1007/s10618-018-0558-x

Kessy A, Lewin A, Strimmer K (2018) Optimal whitening and decorrelation. Am Stat 72(4):309–314. https://doi.org/10.1080/00031305.2016.1277159

Kruskal JB (1964) Nonmetric multidimensional scaling: a numerical method. Psychometrika 29(2):115–129

Lee DW (2001) Genres, registers, text types, domain, and styles: clarifying the concepts and navigating a path through the BNC jungle. Lang Learn Technol 5(3):37–72. https://doi.org/10125/44565

Lijffijt J, Nevalainen T (2017) A simple model for recognizing core genres in the BNC. In: Big and rich data in English Corpus linguistics: methods and explorations (Studies in variation, contacts and change in English 19). University of Helsinki, VARIENG eSeries

Lijffijt J, Papapetrou P, Puolamäki K (2014) A statistical significance testing approach to mining the most informative set of patterns. Data Min Knowl Disc 28(1):238–263. https://doi.org/10.1007/s10618-012-0298-2

Lijffijt J, Kang B, Duivesteijn W, Puolamäki K, Oikarinen E, De Bie T (2018) Subjectively interesting subgroup discovery on real-valued targets. In: 34th IEEE international conference on data engineering, ICDE 2018, Paris, France, April 16–19, 2018, pp 1352–1355. https://doi.org/10.1109/ICDE.2018.00148

Marchini J, Heaton C, Ripley B (2013) fastICA: FastICA algorithms to perform ICA and projection pursuit. R package version 1.2-0. https://CRAN.R-project.org/package=fastICA. Last Accessed: 28 Aug 2019

Paurat D, Gärtner T (2013) InVis: a tool for interactive visual data analysis. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings, Part III, pp 672–676. https://doi.org/10.1007/978-3-642-40994-3_52

Paurat D, Garnett R, Gärtner T (2014) Interactive exploration of larger pattern collections: a case study on a cocktail dataset. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics (IDEA), pp 98–106

Pearson K (1901) On lines and planes of closest fit to systems of points in space. Phil Mag 2(11):559–572

Puolamäki K (2019) sideR—a tool for subjective and interactive visual data exploration in R. https://github.com/edahelsinki/sideR. Last Accessed 28 Aug 2019

Puolamäki K, Papapetrou P, Lijffijt J (2010) Visually controllable data mining methods. In: ICDMW 2010, The 10th IEEE international conference on data mining workshops, Sydney, Australia, 13 December 2010, pp 409–417. https://doi.org/10.1109/ICDMW.2010.141

Puolamäki K, Kang B, Lijffijt J, De Bie T (2016) Interactive visual data exploration with subjective feedback. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2016, Riva del Garda, Italy, September 19–23, 2016, Proceedings, Part II, pp 214–229. https://doi.org/10.1007/978-3-319-46227-1_14

Puolamäki K, Oikarinen E, Kang B, Lijffijt J, De Bie T (2018) Interactive visual data exploration with subjective feedback: an information-theoretic approach. In: 34th IEEE international conference on data engineering, ICDE 2018, Paris, France, April 16–19, 2018, pp 1208–1211. https://doi.org/10.1109/ICDE.2018.00112

R Core Team (2017) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Austria. https://www.R-project.org/. Last Accessed: 28 Aug 2019

Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326. https://doi.org/10.1126/science.290.5500.2323

Ruotsalo T, Jacucci G, Myllymäki P, Kaski S (2015) Interactive intent modeling: information discovery beyond search. Commun ACM 58(1):86–92. https://doi.org/10.1145/2656334

Saeys Y, Van Gassen S, Lambrecht B (2016) Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nat Rev Immunol 16(7):449–462. https://doi.org/10.1038/nri.2016.56

Sedlmair M, Brehmer M, Ingram S, Munzner T (2012) Dimensionality reduction in the wild: gaps and guidance. Technical report TR-2012-03, University of British Colombia, Vancouver

Stahnke J, Dörk M, Müller B, Thom A (2016) Probing projections: interaction techniques for interpreting arrangements and errors of dimensionality reductions. IEEE Trans Visual Comput Graph 22(1):629–638

van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605

van Leeuwen M, Cardinaels L (2015) VIPER—visual pattern explorer. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2015, Porto, Portugal, September 7–11, 2015, Proceedings, Part III, pp 333–336. https://doi.org/10.1007/978-3-319-23461-8_42

Torgerson W (1952) Multidimensional scaling: I. Theory and method. Psychometrika 17(4):401–419

Tukey J (1977) Exploratory data analysis. Behavioral science: quantitative methods. Addison-Wesley, Reading

Venna J, Peltonen J, Nybo K, Aidos H, Kaski S (2010) Information retrieval perspective to nonlinear dimensionality reduction for data visualization. J Mach Learn Res 11:451–490